*Test case*

*f.dat:*

*25 00 54 25 19 25 81 21 02 78*

*19 09 89 77 80 75 91 34 61 24*

*36 42 65 18 81 93 72 34 59 37*

*10 56 17 24 47 02 35 45 19 51*

*26 28 10 23 03 32 65 61 28 95*

*66 63 94 42 77 64 56 80 63 14*

*77 07 34 93 04 19 10 44 76 19*

*86 18 40 47 13 94 98 22 79 94*

*68 37 41 12 06 85 51 85 60 56*

*03 98 29 05 60 15 98 86 04 61*

*77 51 28 24 77 02 36 64 32 05*

*30 73 12 75 14 85 72 60 91 42*

*83 46 03 67 90 48 04 74 41 52*

*62 30 46 71 41 38 80 60 99 05*

*19 48 83 98 11 30 41 72 09 31*

*31 44 21 79 68 97 32 13 62 80*

*61 69 82 25 62 12 83 35 56 19*

*62 74 67 19 41 35 38 16 09 80*

*47 44 85 30 84 53 28 42 07 65*

*99 30 00 91 26 09 91 70 21 14*

*Sample output*

*66 63 94 42 77 64 56 80 63 14  
61 69 82 25 62 12 83 35 56 19  
19 48 83 98 11 30 41 72 09 31  
62 74 67 19 41 35 38 16 09 80  
77 51 28 24 77 02 36 64 32 05  
36 42 65 18 81 93 72 34 59 37  
62 30 46 71 41 38 80 60 99 05  
86 18 40 47 13 94 98 22 79 94  
47 44 85 30 84 53 28 42 07 65  
03 98 29 05 60 15 98 86 04 61  
26 28 10 23 03 32 65 61 28 95  
30 73 12 75 14 85 72 60 91 42  
83 46 03 67 90 48 04 74 41 52  
25 00 54 25 19 25 81 21 02 78  
77 07 34 93 04 19 10 44 76 19  
31 44 21 79 68 97 32 13 62 80  
10 56 17 24 47 02 35 45 19 51  
68 37 41 12 06 85 51 85 60 56  
19 09 89 77 80 75 91 34 61 24  
99 30 00 91 26 09 91 70 21 14*

*Explanation:*

*In test case f.dat:*

*Node.sum\_key: 79 (25+0+54)*

*Node.sum\_key: 117 (19+9+89)*

*Node.sum\_key: 143 (36+42+65)*

*Node.sum\_key: 83 (10+56+17)*

*Node.sum\_key: 64 (26+28+10)*

*Node.sum\_key: 223 (66+63+94)*

*Node.sum\_key: 118 (77+7+34)*

*Node.sum\_key: 144 (86+18+40)*

*Node.sum\_key: 146 (68+37+41)*

*Node.sum\_key: 130 (3+98+29)*

*Node.sum\_key: 156 (77+51+28)*

*Node.sum\_key: 115 (30+73+12)*

*Node.sum\_key: 132 (83+46+3)*

*Node.sum\_key: 138 (62+30+46)*

*Node.sum\_key: 150 (19+48+83)*

*Node.sum\_key: 96 (31+44+21)*

*Node.sum\_key: 212 (61+69+82)*

*Node.sum\_key: 203 (62+74+67)*

*Node.sum\_key: 176 (47+44+85)*

*Node.sum\_key: 129 (99+30+0)*

*After applying the parental node downheap algorithm:*

Node.sum\_key: 223 (66+63+94)

Node.sum\_key: 212 (61+69+82)

Node.sum\_key: 150 (19+48+83)

Node.sum\_key: 203 (62+74+67)

Node.sum\_key: 156 (77+51+28)

Node.sum\_key: 143 (36+42+65)

Node.sum\_key: 138 (62+30+46)

Node.sum\_key: 144 (86+18+40)

Node.sum\_key: 176 (47+44+85)

Node.sum\_key: 130 (03+98+29)

Node.sum\_key: 64 (26+28+10)

Node.sum\_key: 115 (30+73+12)

Node.sum\_key: 132 (83+46+03)

Node.sum\_key: 79 (25+00+54)

Node.sum\_key: 118 (77+07+34)

Node.sum\_key: 96 (31+44+21)

Node.sum\_key: 83 (10+56+17)

Node.sum\_key: 146 (68+37+41)

Node.sum\_key: 117 (19+9+89)

Node.sum\_key: 129 (99+30+0)

Heap keys in array

223, 212, 150, 203, 156, 143, 138, 144, 176, 130, 64,115, 132, 79, 118, 96, 83, 146, 117, 129

Heap keys in tree representation

A diagram of numbers and circles

Description automatically generated